

ABSCHLUSSPRÄSENTATION : FRIDGE-INSIGHT



JAVA PROJEKT :

HAMZA HARTI.

MAHMOUD MATAR.

GITHUB : [HTTPS://GITHUB.COM/FH-ERFURT/FRIDGE-INSIGHT.GIT](https://github.com/fh-erfurt/fridge-insight.git)

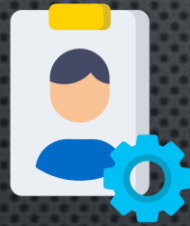
PROJEKT BESCHREIBUNG

- **FRIDGE-INSIGHT** IST EINE SOFTWARE, MIT DER SIE IHREN KÜHLSCHRANK BESSER VERWALTEN KÖNNEN. ES ENTHÄLT VIELE GUT PROGRAMMIERTE FUNKTIONEN, UM EIN OPTIMALES VERWALTUNGSSYSTEM ZU GEWÄHRLEISTEN.

MUSS-KRITERIEN

✓ BENUTZER VERWALTEN.

- KONTO ADDIEREN
- KONTO BEARBEITEN
- KONTO ENTFERNEN



✓ EINKAUF LISTE VERWALTEN



- EINKAUFSKORB ERSTELLEN
- ELEMENT ZUM EINKAUFSKORB HINZUFÜGEN
- ELEMENT VOM EINKAUFSKORB ENTFERNEN

✓ BENACHRICHTIGUNGEN KONFIGURIEREN

- BENACHRICHTIGUNGEN ERSTELLEN
- BENACHRICHTIGUNGEN BEARBEITEN
- BENACHRICHTIGUNGEN ENTFERNEN



✓ HINZUFÜGEN/ENTFERNEN EINES ELEMENTS

- PRODUKT ADDIEREN
- PRODUKT BEARBEITEN
- PRODUKT ENTFERNEN



WUNSCH-KRITERIEN

- ✓ DIE MENGE DER LEBENSMITTEL SOLLTE AUTOMATISCH AKTUALISIERT WERDEN.
- ✓ DIE BENACHRICHTIGUNGEN SOLLEN BEI LEBENSMITTELKNAPPHEIT AUTOMATISCH ERSTELLT WERDEN.
- ✓ JEDER BENUTZER HAT DIE MÖGLICHKEIT, EIN PROFIL ZU ERSTELLEN, IN DEM VIELE FUNKTIONEN ZUR VERFÜGUNG STEHEN.

ABGRENZUNGS-KRITERIEN

- ✓ NUR DER "SUPERUSER" HAT DIE BERECHTIGUNG, EIN KONTO HINZUZUFÜGEN, ZU ÄNDERN ODER ZU ENTFERNEN.
- ✓ NUR EIN "SUPERUSER" KANN EIN PRODUKT HINZUFÜGEN, ÄNDERN ODER LÖSCHEN
- ✓ NUR EIN "SUPERUSER" KANN EIN ANDERES KONTO ALS "SUPERUSER" KENNZEICHNEN.

AKTEURE: BERECHTIGUNGEN UND BEGRENZUNGEN

Rolle	Beschreibung	Vertreter(Bsp.)	Bemerkungen
Super User	Der Superuser kann auf alle Funktionen des Programms zugreifen.	Vater/Mütter	
User	Der User kann auf alle grundlegenden Funktionen zugreifen	Kinder/Freunde	Vertreter kann Super User werden, aber er braucht die Erlaubnis eines der Super Users

Benötigst du ein gültiges Element?



Benötigt gleichzeitige Verwendung eines Elements



Person	
firstName	String
lastName	String
Age	int
Height	float
superUser	boolean
familyPosition	String
Person()	
Person(String, String, int, float, boolean, String)	
getFirstName()	String
getLastName()	String
getFamilyPosition()	String
isSuperUser()	boolean
getHeight()	float
getAge()	int
setAge(int)	void
setFamilyPosition(String)	void
setFirstName(String)	void
setHeight(float)	void
setLastName(String)	void
setSuperUser(boolean)	void

Food	
foodTitle	String
expireDate	String
foodType	String
foodQuantity	int
foodMarket	SuperMarket
Food()	
Food(String, String, String, int, SuperMarket)	
getFoodTitle()	String
getFoodType()	String
getExpireDate()	String
getFoodQuantity()	int
getFoodMarket()	SuperMarket
setExpireDate(String)	void
setFoodMarket(SuperMarket)	void
setFoodQuantity(int)	void
setFoodTitle(String)	void
setFoodType(String)	void

Address	
houseNumber	int
streetName	String
City	String
State	String
postalCode	int
Address()	
Address(int, String, String, String, int)	
getHouseNumber()	int
getPostalCode()	int
getCity()	String
getState()	String
getStreetName()	String
setCity(String)	void
setHouseNumber(int)	void
setPostalCode(int)	void
setState(String)	void
setStreetName(String)	void

UserList	
Users	LinkedList<Account>
UserList()	
getUsers()	
indexOfElement(Account)	
getElementByIndex(int)	
setUsers(LinkedList<Account>)	
addUserToList(Account)	
superUserAdd(Account, Account)	
deleteUserFromList(String, String, String)	
superUserDelete(String, String, Account)	
getUser(String, String)	
updateUser(String, String, int, float, String, String, String, String, String) void	

Account	
dateOfCreation	String
Password	String
Account()	
Account(String, String, int, float, boolean, String, String, String)	
setDateOfCreation(String)	
getDateOfCreation()	
getPassword()	
setPassword(String)	

PurchaseList	
listMaker	Account
dateOfPurchase	String
foodToBePurchased	LinkedList<Food>
PurchaseList()	
PurchaseList(Account, String, LinkedList<Food>)	
getFoodToBePurchased()	LinkedList<Food>
getListMaker()	Account
getDateOfPurchase()	String
setDateOfPurchase(String)	void
setListMaker(Account)	void
setFoodToBePurchased(LinkedList<Food>)	void

NotificationFood	
minimumQuantity	int
selectedFood	Food
dateOfExpiration	String
NotificationFood()	
NotificationFood(int, Food, String)	
getSelectedFood()	Food
getMinimumQuantity()	int
setMinimumQuantity(int)	void
setSelectedFood(Food)	void
getDateOfExpiration()	String
setDateOfExpiration(String)	void

AddressTest	
Address()	void
getHouseNumber()	void
getPostalCode()	void
getCity()	void
getState()	void
getStreetName()	void
setCity()	void
setHouseNumber()	void
setPostalCode()	void
setState()	void
setStreetName()	void

FoodTest	
food()	void
setExpireDate()	void
setFoodMarket()	void
setFoodQuantity()	void
setFoodTitle()	void
setFoodType()	void
getFoodTitle()	void
getExpireDate()	void
getFoodQuantity()	void
getFoodMarket()	void

ShoppingCart	
shoppingCart	LinkedList<PurchaseList>
ShoppingCart()	
getShoppingCart()	
setShoppingCart(LinkedList<PurchaseList>)	
addPurchaseListToCart(PurchaseList)	
getPurchaseListFromCart(PurchaseList)	
deletePurchaseListFromCart(PurchaseList)	
updatePurchaseListInCart(PurchaseList, Account, String, LinkedList<Food>) void	

NotificationFoodList	
Notifications	LinkedList<NotificationFood>
NotificationFoodList()	
getNotifications()	
setNotifications(LinkedList<NotificationFood>)	
addNotificationToList(NotificationFood)	
deleteNotificationFromList(NotificationFood)	
updateNotificationInList(NotificationFood, String, int, Food) void	
getNotificationFromList(NotificationFood)	

SuperMarket	
marketName	String
marketAddress	Address
SuperMarket()	
SuperMarket(String, Address)	
getMarketAddress()	Address
getMarketName()	String
setMarketAddress(Address) void	
setMarketName(String) void	

UserListTest	
getUsers()	void
setUsers()	void
addUserToList()	void
superUserAdd()	void
deleteUserFromList()	void
superUserDelete()	void
getUser()	void
updateUser()	void

PurchaseListTest	
PurchaseList()	void
setDateOfPurchase()	void
setListMaker()	void
setFoodToBePurchased()	void
getFoodToBePurchased()	void
getListMaker()	void
getDateOfPurchase()	void

NotificationFoodTest	
NotificationFood()	void
setMinimumQuantity()	void
setSelectedFood()	void
setDateOfExpiration()	void
getSelectedFood()	void
getMinimumQuantity()	void
getDateOfExpiration()	void

ShoppingCartTest	
getShoppingCart()	void
setShoppingCart()	void
addPurchaseListToCart()	void
getPurchaseListFromCart()	void
deletePurchaseListFromCart()	void
updatePurchaseListInCart()	void

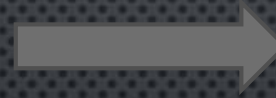
NotificationFoodListTest	
getNotifications()	void
setNotifications()	void
addNotificationToList()	void
deleteNotificationFromList()	void
updateNotificationInList()	void
getNotificationFromList()	void

AccountTest	
setDateOfCreation()	void
getDateOfCreation()	void
getPassword()	void
setPassword()	void
Account()	void

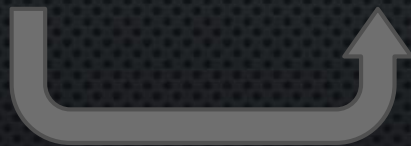
SuperMarketTest	
SuperMarket()	void
getMarketAddress()	void
getMarketName()	void
setMarketAddress()	void
setMarketName()	void

EINKAUFSKORB VERWALTUNG

PurchaseList		
f	listMaker	Account
f	dateOfPurchase	String
f	foodToBePurchased	LinkedList<Food>
m	PurchaseList()	
m	PurchaseList(Account, String, LinkedList<Food>)	
m	getFoodToBePurchased()	LinkedList<Food>
m	getListMaker()	Account
m	getDateOfPurchase()	String
m	setDateOfPurchase(String)	void
m	setListMaker(Account)	void
m	setFoodToBePurchased(LinkedList<Food>)	void



ShoppingCart		
f	shoppingCart	LinkedList<PurchaseList>
m	ShoppingCart()	
m	getShoppingCart()	LinkedList<PurchaseList>
m	setShoppingCart(LinkedList<PurchaseList>)	void
m	addPurchaseListToCart(PurchaseList)	void
m	getPurchaseListFromCart(PurchaseList)	PurchaseList
m	deletePurchaseListFromCart(PurchaseList)	void
m	updatePurchaseListInCart(PurchaseList, Account, String, LinkedList<Food>)	void



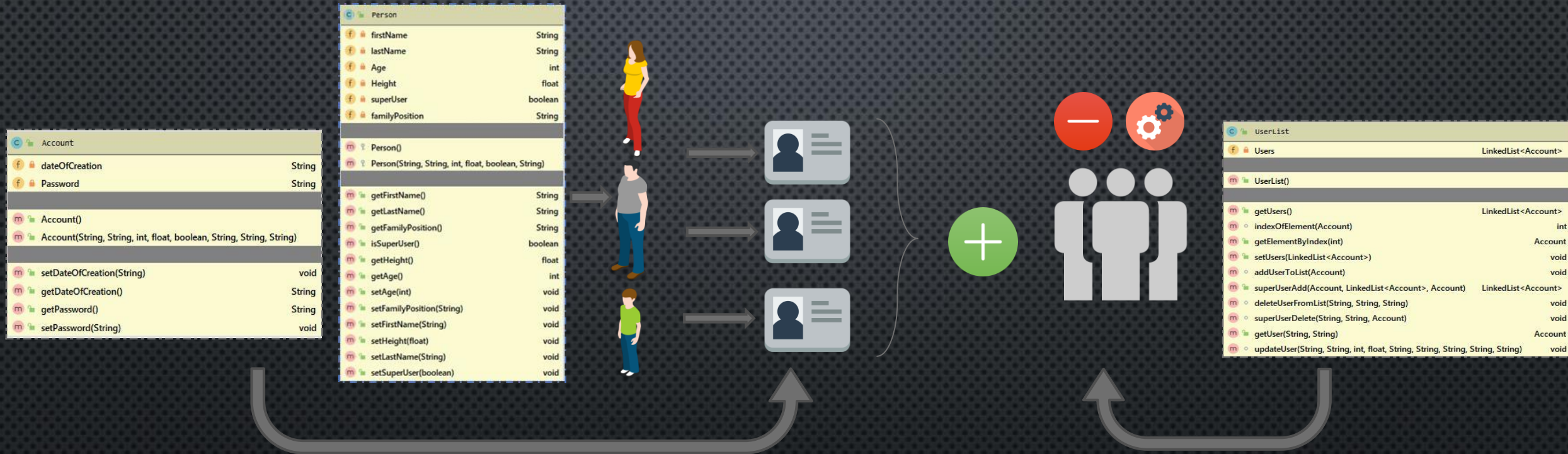
BENACHRICHTIGUNGEN KONFIGURIEREN

	NotificationFood	
	minimumQuantity	int
	selectedFood	Food
	dateOfExpiration	String
	NotificationFood()	
	NotificationFood(int, Food, String)	
	getSelectedFood()	Food
	getMinimumQuantity()	int
	setMinimumQuantity(int)	void
	setSelectedFood(Food)	void
	getDateOfExpiration()	String
	setDateOfExpiration(String)	void



	NotificationFoodList	
	Notifications	LinkedList<NotificationFood>
	NotificationFoodList()	
	getNotifications()	LinkedList<NotificationFood>
	setNotifications(LinkedList<NotificationFood>)	void
	addNotificationToList(NotificationFood)	void
	deleteNotificationFromList(NotificationFood)	void
	updateNotificationInList(NotificationFood, String, int, Food)	void
	getNotificationFromList(NotificationFood)	NotificationFood

BENUTZER VERWALTEN



LESSONS LEARNED

- MIT STRESS BESSER UMGEHEN.
- TESTS ERSTELLEN UND AUSFÜHREN.
- PLANNUNG UND KOORDINIERUNG IST SEHR WICHTIG.
- BESSERE MANAGEMENTFÄHIGKEITEN.

VIELEN DANK FÜR DIE AUFMERKSAMKEIT